

ONSET: Automated Foundational Ontology Selection and Explanation

Zubeida Khan and C. Maria Keet

School of Mathematics, Statistics, and Computer Science, University of KwaZulu-Natal and UKZN/CSIR-Meraka Centre for Artificial Intelligence Research, South Africa, zkhan@csir.co.za, keet@ukzn.ac.za

Abstract. It has been shown that using a foundational ontology for domain ontology development is beneficial in theory and practice. However, developers have difficulty with choosing the appropriate foundational ontology, and why. In order to solve this problem, a comprehensive set of criteria that influence foundational ontology selection has been compiled and the values for each parameter determined for DOLCE, BFO, GFO, and SUMO. This paper-based analysis is transformed into an easily extensible algorithm and implemented in the novel tool ONSET, which helps a domain ontology developer to choose a foundational ontology through interactive selection of preferences and scaling of importance so that it computes the most suitable foundational ontology for the domain ontology and explains why this selection was made. This has been evaluated in an experiment with novice modellers, which showed that ONSET greatly assists in foundational ontology selection.

1 Introduction

Ontology development and usage is increasing, which puts higher demands on sound ontology engineering guidelines and practices. One component of ontology development may involve selection and usage of a foundational ontology (FO) that provides high-level categories and generic relationships that are common among ontologies. If used, it increases the domain ontology's quality and interoperability, and it has been shown to speed up ontology development [1,2]. Although there are a variety of FOs available since several years, such as DOLCE [3], BFO, GFO [4], SUMO [5] and YAMATO, most existing methodologies, such as METHONTOLOGY, NeOn, On-To-Knowledge, and the MeltingPoint methodology, do not include their usage explicitly, whereas OntoSpec [6] contains specific guidelines on how to use DOLCE only. Even when a domain ontology developer wants to consider using a FO, there is a prohibitive learning curve due to the considerable quantity of documentation and the new terminology it introduces. Seeing that FOs are beneficial and sometimes necessary in domain ontology development and that there is no methodology or tools which consider FOs in general, there is a need for assistance. We focus on the essential step before including the use of a FO in an extended ontology development methodology: a method for selecting the appropriate FO for the existing or prospective

domain ontology and being able to explain why that one is, comparatively, the best choice. To date, there have been a few partial, paper-based, comparisons of the ontologies tailored to an individual project only [7,8,9], but not such that it can be used regardless the scenario and subject domain.

We aim to fill this gap by providing *software-supported selection* of the most suitable FO based on the developer’s input, and therewith also generate an *automated explanation* based on the FO’s features and the input previously provided by the developer. To realise this, we analysed the characteristics of the DOLCE, BFO, GFO, and SUMO FOs, such as their philosophical distinctions, and availability in certain ontology languages, and on what criteria domain ontology developers have based their decisions in existing projects. A first, extensible, selection and explanation algorithm was developed and implemented in the Ontology Selection and Evaluation Tool ONSET handling DOLCE and BFO only, and its extensibility was evaluated with relative ease of adding the values for the criteria of GFO and SUMO, and corresponding explanations. In addition, to assist the developer in commencing using the FO, ONSET lists relevant references of existing ontologies that also used the proposed FO. This version of ONSET was evaluated with existing ontology development projects and in a controlled experiment with novice ontology developers. Those who used ONSET selected the appropriate FO in most cases and, unlike the null group, they did it quicker and they were able to provide much more reasons for selecting the FO.

Ontology recommender systems are available (e.g., [10,11,12]), but they concern guidance in finding suitable domain ontologies, which is a different scenario for which other questions and requirements hold than for selecting a FO. They share the idea of assisting and informing developers about the criteria and properties associated with ontologies, but ONSET also provides explanation why some FO suits better than another—therewith also providing a basis for argumented comparison of the FOs—, how it relates to the domain ontology to be created, and relevant references of ontologies that also used that FO. ONSET and supplementary material can be accessed from <http://www.meteck.org/files/onset/>.

The remainder of the paper is structured as follows. A brief methodology is described in Section 2, which is followed by the paper-based comparison in Section 3. The design and implementation of ONSET is presented in Section 4 and its evaluation with results and discussion in Section 5. We conclude in Section 6.

2 Methodology

To solve the problems and realise ONSET, we followed a standard software development methodology, and adapted it to the problem at hand:

1. Conduct a literature review on the usage of FOs.
2. Carry out a comparative study of popular FOs.
3. Select suitable FOs to implement in the tool
4. Create an initial list of criteria to assess the ontologies on and, hence, on why one would use either ontology.

5. Contact the creators of the selected ontologies to verify and contribute to the initial criteria list.
6. Devise tool requirements.
7. Based on these criteria, produce an algorithm in order to assist the user in development.
8. Design and implement the algorithm in an easily extendable application.
9. Perform a qualitative evaluation of the tool by FO usage scenarios
10. Test extensibility of the approach by including another FO and analyse changes required, if any.
11. Perform a qualitative evaluation of the tool by FO usage scenarios and a quantitative evaluation with novice ontology developers.

3 Foundational ontologies and their comparison

We conducted a literature review of about 60 papers, covering the publications and documentation by the FO's developers, other comparisons, and case reports on actual usage in domain ontologies, of which we discuss a notable selection.

3.1 Related works

The WonderWeb deliverable D18 [3] contains extensive information on DOLCE, OCHRE and BFO v1. The Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) is based on common-sense principles, heavily informed by Ontology, and axiomatised in FOL. The Basic Formal Ontology (BFO) is a small taxonomy commonly used for scientific research and data integration purposes. The Suggested Upper Merged Ontology (SUMO) [5] is a descriptive ontology of universals and particulars, developed within the context of the IEEE-sanctioned working group and having merged several ontologies. The General Formal Ontology (GFO) [4] is an ontology of universals, concepts and symbols, and is used mainly in the health-care and medical fields.

There are several comparisons of FOs [9,13,14,15], which focus on selected topics, such as technical aspects and representation languages, which can be extended with more recent topics for comparison, such as modularity. However, these comparative studies did not offer a range of categories to compare. Ontological commitments, ontology languages, subject domain examples and other categories were all grouped and compared together to suit the one-off scenario. Furthermore, these comparisons used only a subset of relevant criteria to substantiate usage of a particular FO.

The areas of the domain ontologies using a FO are diverse, yet some differences can be observed. Scientific ontologies such as those used in the biomedical and life science domains mainly use BFO and GFO [16,17] which is partially due to the OBO Foundry [18], which has recommended that ontologies registered on the OBO Foundry use BFO. DOLCE and SUMO have been applied to a variety of subject domains including engineering [14], biomedical [19,20], government and military [15], and landscape [21].

3.2 Categories of criteria and comparison

By analysing the comparative studies, documentation of FOs, and their usage in domain ontology projects, an initial list of criteria for selection was created, and grouped into five categories, which are most relevant for the ‘information systems perspective’ of ontology development (cf. choosing, e.g., which mereology and axioms to incorporate in the ontology [16,22]):

- *Ontological Commitments*: The philosophical choices taken by FOs; e.g., an ontology of particulars vs. universals, multiplicative or not.
- *Representation Language*: The languages used to represent an ontology; e.g., KIF, OBO, or OWL DL.
- *Software engineering properties*: General properties associated with FOs; e.g., licensing and whether the FO is modularised.
- *Subject Domain*: Existing domains represented in a domain ontology using a FO; e.g., the biomedical domain.
- *Applications*: The application scenarios of domain ontologies; e.g., whether the intended purpose of the domain ontology is for the Semantic Web, data integration, NLP.

The final lists of criteria for the ontological commitments and their corresponding values for the four selected FOs is shown in Table 1. Note some idiosyncracies in terminology usage; e.g., an ‘ontology of particulars’ may well focus on adding classes to the high-level categories provided in the FO and while ‘descriptive’ and ‘realist’ is important philosophically, one can include them both in different sections of an ontology (in GFO; or, from a logicians’ viewpoint: it all ends up as a logical theory anyway). The final lists of criteria for ontology representation languages for the four selected FOs is as follows. DOLCE is available in FOL, KIF, OWL DL (and therewith also OWL 2 DL), BFO is available in OBO, FOL, KIF, and all OWL species, GFO in OWL DL (and also OWL 2 DL), and SUMO in SUO-KIF and OWL DL. Software engineering properties as per the final criteria lists are compared in Table 2 for their dimensions and modularity; regarding licencing: they are all freely available, and they are all actively being maintained. The complete list of subject domain and application criteria can be accessed at <http://www.meteck.org/files/onset/>.

4 Design of ONSET

A systematic and rigorous approach is employed for the design of the ONtology Selection and Evaluation Tool ONSET in order to ensure proper functioning and useful functionality. The design concerns the tool’s requirements, selection algorithm, and implementation aspects.

4.1 Requirements for ONSET

The tool has a number of functional requirements. Primarily, it must select an appropriate FO to be used, and a neat summary of why the particular FO was selected is to be produced as an output (hence, it also must store a user’s answers

Table 1. Comparison of ontological commitments

Term (and very brief descriptions of its meaning)	DOLCE	BFO	GFO	SUMO
Universals vs. Particulars (Universals can have instances, particulars do not)	Particulars	Universals	Universals, concepts, and symbols	Universals and particulars
Descriptive vs. Realist (Descriptive: represent the entities underlying natural language and human common-sense; Realist: represent the world exactly as is)	Descriptive	Realist	Descriptive and Realist	Descriptive
Multiplicative vs. Reductionist (Multiplicative: different objects can be co-located at the same time; Reductionist: only one object may be located at the same region at one time)	Multiplicative	Reductionist	Unclear	Multiplicative
Endurantism vs. Perdurantism (Endurantism: an object is wholly present at all times; Perdurantism: an object has temporal parts)	Endurantism and perdurantism	Endurantism and perdurantism	Endurantism and perdurantism	Endurantism and perdurantism
Actualism vs. Possibilism (everything that exists in the ontology is real vs. objects are allowed independent of their actual existence)	Possibilism	Actualism	Unclear	Unclear
Eternalist stance (the past, present and future all exist)	Eternalist stance	Eternalist stance	Eternalist stance	Eternalist stance
Concrete & Abstract entities (Concrete: entities that exist in space and time; Abstract: entities that exist neither in space nor time)	Concrete, abstract	Concrete	Concrete, abstract	Concrete, abstract
Mereology (theory of parts)	GEM	Own mereology	Own mereology	Own mereology
Temporal aspects (e.g., time-indexed axioms)	Provided	Not provided	Provided	Provided
Granularity (different levels of detail contained in an ontology)	High level	Sensitive to granularity	Unclear	Unclear
Properties and values ('attribute'; e.g., the colour of an apple)	Included	Not included	Included	Included
Model for space and time (Consists of time and space regions and boundaries)	Not included	Not included	Included	Not included
One-layered vs. Three-layered architecture (a basic level only; an abstract top level, abstract core level and basic level)	One-layered	One-layered	Three-layered architecture	One-layered
Situations and situoids (Situation: an aggregate of facts that can be comprehended as a whole and satisfies certain conditions of unity; Situoid: is a part of the world that is a comprehensible whole and can exist independently)	Not included	Not included	Included	Not included

Table 2. Comparison of 2 of the 4 software engineering properties.

	Dimensions	Modularity
DOLCE	100 categories and 100 axioms + relations, quality properties and qualia to represent attributes	Lighter/expressive versions, endurants and perdurants are separate, built-in domain-specific ontologies
BFO	in OWL - 39 universals; in OBO- 23 terms and 33 typedefs; with RO-33 universals and 34 object properties	Endurants and perdurants are separate
GFO	Full- 79 classes, 97 subclass axioms and 67 object properties; Basic- 44 classes, 28 subclass axioms, 41 object properties	Lighter/expressive versions, modules for functions and roles
SUMO	1000 terms, 4000 axioms, 750 rules	Endurants and perdurants separate, built-in domain-specific ontologies

corresponding to each question). If the user has requirements relating to more than one FO, the conflicting results—what is provided by the selected FO compared to what the user wants—is to be compared and displayed. In addition, it has to provide a list of existing ontology references of the domain chosen by the user, if available. It must include ‘additional questions’, when applicable: these are the questions where all implemented FOs have the same value and thus will not affect the results of ONSET at present, but may in the future, and the user must be given a choice about whether to include or exclude these questions from the program run. Optional scaling, which involves assigning a rating of importance to each category by the user, must be implemented.

In addition, several non-functional requirements are essential. The tool must be designed and implemented such that maintaining and modifying it is a quick and simple process and it must be able to run on different operating systems and platforms. Users must feel comfortable and at ease using the tool. The tool should be divided into windows, tabs and panels to minimize possible cognitive overload and make it easy to understand and follow, and promote usability. The time taken between submitting answers and calculating results must be minimal.

4.2 Algorithm

The general idea of the algorithm is that it takes the largest counter variable to select the FO and it uses arrays of ontological choices to display the motivation for the selected FO and conflicting results. Algorithm 1 decides whether additional questions are to be shown in the interface (lines 1-5), and assigns an optional scaling per category (lines 6-10), both according to the user’s input. The algorithm then applies the selected scaling values to each category (line 19). It displays questions and options per category (lines 16,17), and accepts and stores the answers of the user (line 20). Based on this, in Algorithm 2, the selected FO is calculated and displayed, alongside reasons as to why it was chosen (lines 1-5). If present, conflicting results are displayed (lines 8-21). In addition, it provides a list of existing ontology references of the domain chosen by the user, if available

(lines 6,7). The algorithm is easily extensible in the sense that, when a new FO has to be added, the structure and data contained in the algorithm change little: if all the (manually) identified criteria for using such a foundational already exists in ONSET's repository, only references to subject domains using that FO have to be added, and if there is some criterion that does not exist in ONSET yet, a question and options are to be added.

Algorithm 1: ONtology Selection and Evaluation Tool Algorithm 1

```

DolceCount = 0; BFOCount = 0; GFOCount = 0; SUMOCCount = 0;
DolceAnswers[] = null; BFOAnswers[] = null; GFOAnswers[] = null;
SUMOAnswers[] = null; DOLCEdomain[] = null; BFOdomain[] = null;
GFOdomain[] = null; SUMODomain[] = null; k = 0; ScalingValues[] = null;
output: Include additional questions?
1 if input is yes then
2   | Show additional questions
3 else
4   | Hide additional questions
5 end
   output: Assign scaling per category?
6 if input is yes then
7   | for i ← 0 to numOfCategories do
8     |   Read scaling value;
9     |   Store scaling value in ScalingValues[i];
10    | end
11 else
12   | ScalingValues[i] = 1;
13 end
14 for i ← 0 to numOfCategories do
15   | for j ← 0 to numOfQuestionsPerCategory do
16     |   Display question;
17     |   Display options;
18     | if option corresponds to DOLCE then
19       |   DOLCECount = DOLCECount+(1* ScalingValues[i]);
20       |   DOLCEANSWERS[j]= option text;
21       | if numOfCategories == 3 then
22         |   for k ← 0 to numberofreferences do
23           |   | DOLCEdomain[k] = Subject domain reference;
24           |   end
25         |   end
26       | else if option corresponds to BFO then
27         |   %% analogous to lines 18-25
28       | else if option corresponds to GFO then
29         |   %% analogous to lines 18-25
30       | else if option corresponds to SUMO then
31         |   %% analogous to lines 18-25
32       |   end
33     | end
34 end

```

Algorithm 2: Ontology Selection and Evaluation Tool Algorithm 2

```
input : Calculate result
1 if DOLCECount > (BFOCount AND GFOCount AND SUMOCCount) then
  output: Selected FO is FO
  output: Reasons why DOLCE was chosen:
  2 for  $i \leftarrow 0$  to DOLCEAnswers.length do
    3   if DOLCEAnswers[ $i$ ]! = null then
      | output: DOLCEAnswers[ $i$ ]
    4   end
  5 end
  output: Existing ontologies with the specified subject domain
  6 for  $k \leftarrow 0$  to DOLCEDomain.length do
    | output: DOLCEDomain[ $k$ ]
  7 end
  8 if BFOAnswers OR GFOAnswers OR SUMO not empty then
    output: Conflicting results: The tool has detected that some of your
      criteria matches with other foundational ontologies.
    9   for  $i \leftarrow 0$  to BFOAnswers.length do
      10   | if BFOAnswers[ $i$ ]! = null then
        | | output: BFOAnswers[ $i$ ]
      11   | end
    12   end
    13   for  $i \leftarrow 0$  to GFOAnswers.length do
      14   | if GFOAnswers[ $i$ ]! = null then
        | | output: GFOAnswers[ $i$ ]
      15   | end
    16   end
    17   for  $i \leftarrow 0$  to SUMOAnswers.length do
      18   | if SUMOAnswers[ $i$ ]! = null then
        | | output: SUMOAnswers[ $i$ ]
      19   | end
    20   end
    21 end
  22 else if BFOCount > (DOLCECount AND GFOCount AND SUMOCCount)
    then
  23   | %% analogous to lines 1-21
  24 else if GFOCount > (DOLCECount AND BFOCount AND SUMOCCount)
    then
  25   | %% analogous to lines 1-21
  26 else if SUMOCCount > (DOLCECount AND BFOCount AND GFOCount)
    then
  27   | %% analogous to lines 1-21
  28 else
    output: :
  29   The tool was not able to select a FO due to many a contradictory responses.
    Restart the process if you wish.
  30 end
```

4.3 Implementation Specification

ONSET was developed in Java using Netbeans IDE. The .jar file that runs locally on the user's machine requires a minimum of 1MB of free disk space, 1GB of RAM, and JRE components installed.

The functional requirements are met as follows. All five categories with their criteria are converted into questions with pre-defined answers that a user can choose from. A simple if-else statement handles the optional additional questions. For each category, a scaling value between 0-5 is available where 0 represents omit and 5 represents most important, which is applied to each answered question. Answers are stored in an array, to which the calculations are applied, resulting in the ontology with the highest value being selected, and the array of the preferences and values of the selected FO is displayed to the user. Likewise, if the array of the FO that was not selected is not empty, conflicting answers are found, which are displayed alongside what is offered by the selected FO, offering a comparison to the user. Finally, when a user chooses a subject domain, all the available references are added to an array of subject domains and displayed on the results page.

The non-functional requirements are met as follows. Portability is achieved by having a .jar file produced by Netbeans IDE, which is platform-independent, and maintainability is achieved by means of useful comments throughout the code and generating java docs. Response time during usage of ONSET is minimal thanks to the non-frills, uniform, and neat design of the tool. Finally, usability was enhanced by helpful labels and "explain" buttons for explaining complicated terms, which are slightly longer than the explanations in Table 1.

5 Results and Discussion

ONSET was evaluated in three ways: (i) testing it against existing and simulated ontologies, (ii) a quantitative evaluation with novice ontology developers, and (iii) qualitative, expert user feedback from peers.

5.1 Evaluation of ONSET's functionality

We illustrate ONSET with existing and simulated ontologies; more scenarios can be found at <http://www.meteck.org/files/onset/>.

Scenario 1: Semantic Management of Middleware. The tool was tested according to the requirements of [9] which is an application of the semantic web. Ontological choices of the test case include: descriptiveness, a multiplicative approach, possibilism, perdurantism, modularity (lightweight versions) and an executable language. An example of one of the questions in ONSET which corresponds to this scenario is "Descriptive or Realist Ontology?", where the user has to choose between "Descriptive", "Realist" and "Both". When all the choices are submitted to ONSET, it chooses DOLCE as a FO (see Fig. 1). This corresponds to the foundational ontology used in [9].

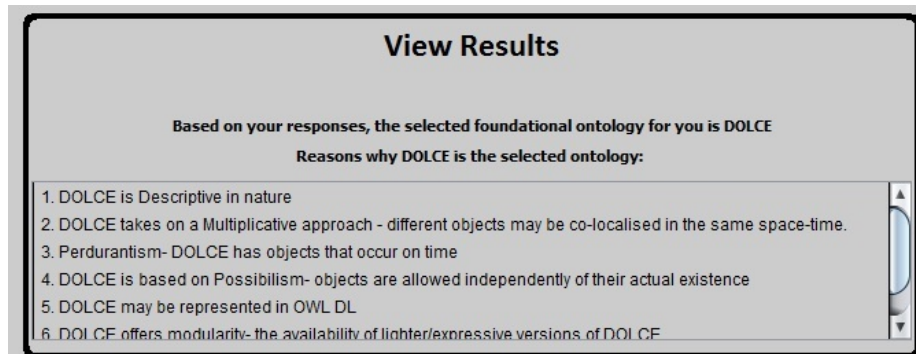


Fig. 1. Output of ONSET: Scenario 1

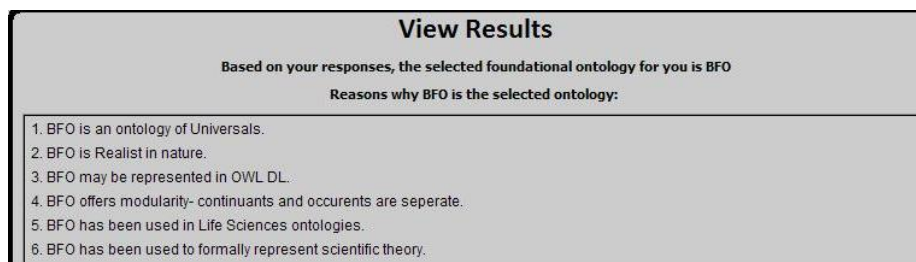


Fig. 2. Output of ONSET: Scenario 2 (without scaling).

Scenario 2: Scaling effects. We show the functioning of scaling in ONSET by simulation of a scenario. Let us assume that there is an ontology to be created with the following requirements: an ontology of universals, realist in nature, to be represented in OWL DL, modularity (endurants and perdurants separate, built-in domain specific ontologies), applying it to formally represent a scientific theory and a domain of life sciences. Without scaling, ONSET chooses BFO as the selected FO as can be seen in Fig. 2. We then use the same input but fill in priorities for the categories in the scaling section in ONSET; for instance, we assign ontological commitments a value of 1, representation languages 5, software engineering properties 3, subject domain 5, and applications a 4. Based on the results calculated, ONSET now chooses DOLCE as the selected FO (Fig. 3); hence, the results of ONSET changed for the same values as before but together with scaling. Observe also the reporting of conflicting answers in Fig. 3.

5.2 Experimental evaluation

The purpose of the quantitative evaluation is to assess whether using ONSET makes a difference in selecting a FO compared to not using the tool, focusing on timing and correctness and completeness of the selections.

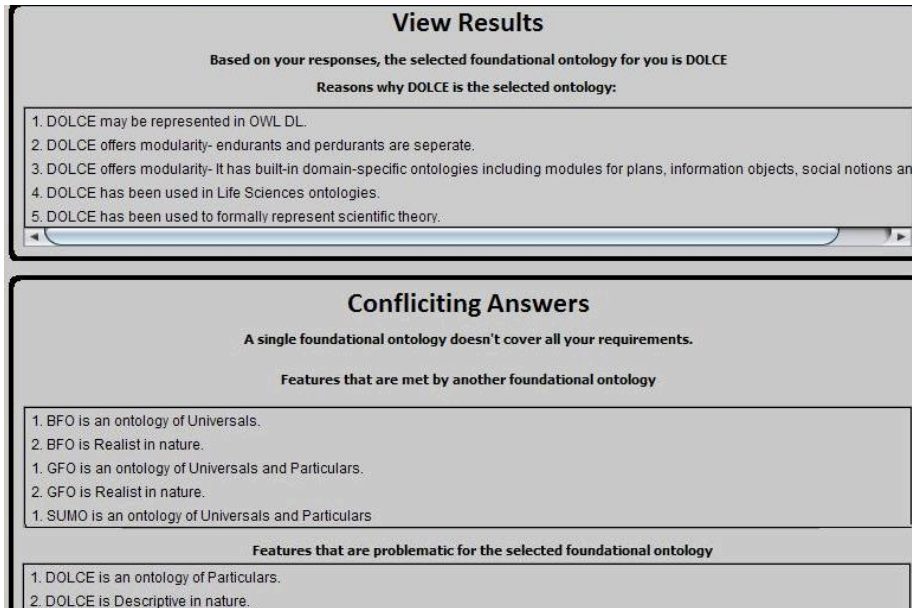


Fig. 3. Output of ONSET: Scenario 2 (with scaling).

Materials and methods. The set-up for the experiment is as follows:

1. Lecture on FOs (1.5h) and announcement of experiment to be held one week after the lecture during the lab time.
2. Divide class into groups A and B randomly (list generated by random.org).
3. Explain purpose of experiment and distribute the assessment to everyone.
4. Provide both the groups with instructions (soft-copy):
 - Group A: complete the given tasks in the prescribed time (≤ 2 h) to the best of their ability by using their lecture notes and resources found on the internet.
 - Group B: idem as Group A, and using ONSET.

The tasks consisted of (i) five scenarios for domain ontology development, where for each scenario, a domain ontology had to be chosen and reasons given why, (ii) open-ended questions asking their opinion on the experiment, and (iii) for Group B only, feedback on the tool.

5. Participants upload their final answers to the course's Moodle, so that timing is captured.
6. Evaluate the tasks, which is performed as follows:
 - i. Assessing and comparing the quality of answers of group A and B.
 - ii. Comparing the time taken to complete the tasks of each group.
 - iii. Collecting and analysing user opinions of all participants.

To measure the quality of the answers given by the participants, we use an accuracy measure: one mark is awarded if the FO corresponds to that of the scenario, and thereafter one mark is awarded for each correct reason provided

for the scenario (‘correct’ with respect to the scenario). Then, given a maximum amount of criteria for each question, the marks for all the scenarios are summed and converted to a percentage.

Results and discussion. The sample size was 18 honours (4th year) computer science students enrolled in UKZN’s “ontologies and knowledge bases” course (comp718), with 9 students in group A and 9 in group B.

Group A (the null group) had submitted their tasks in, on average, 105 minutes and of the $(9 * 5 =)$ 45 scenarios, 8 scenarios were left unanswered. For each scenario, the participants had experienced difficulty in selecting a FO, and could not substantiate their choices. 67% of the participants stated that they had found it difficult to identify which FO would satisfy the criteria provided. Some participants thought that substantiating their choices for FO selection was time-consuming, and had difficulty in understanding all the ontological terms used.

Group B (using ONSET) had submitted their tasks in, on average, 97 minutes with all scenarios answered completely. Thus, the time taken for FO selection is somewhat less in Group B than in Group A. They were able to select a FO to use and provide solid reasons for their choice. The participants of Group B all had a good understanding of the ontological terms used, which, to a large extent, is because ONSET provides “explain” buttons. 33% of the participants of Group B stated that ONSET assisted with complex terms and concepts.

The accuracy rates for each task and of the total tasks are included in Table 3, from which can be seen that for each use-case, Group B was more than twice as accurate as Group A in FO selection for each scenario, and has about 3 times higher accuracy percentage overall. The largest difference in accuracy rates between the two groups is found in scenario 5. Scenario 5 is a complex problem involving the representation of more abstract knowledge. The participants of Group A struggled to identify a FO that would allow this. Group B, on the other hand, found it much easier because by answering with the correct criteria, ONSET selected the appropriate FO, with explanations. Overall, though, the accuracy rates of Group B are not as high as one may hope for, which is mainly because the participants were not able to provide every possible reason for selecting a FO. This is probably because it was their first exercise in working with such scenarios. Notwithstanding, the accuracy rates of Group B are much higher than that of Group A. Thus, overall, Group B performed better in FO

Table 3. A comparison of the accuracy of the answers by Group A and Group B.

Scenario	Group A Average	Group B Average
1.Ontology of heart diseases	22%	52%
2.Ontology for the integration of databases of a manufacturing factory	16%	43%
3.Ontology of economic systems	20%	48%
4.Ontology of banks	16%	37%
5.Ontology for conceptual data models	8%	51%
<i>All Scenarios</i>	<i>16%</i>	<i>46%</i>

selection than Group A and it is apparent that ONSET does provide assistance in FO selection.

Qualitative feedback Participants of Group B were impressed with the output of ONSET, and thought that it generates results effectively, provided that you input sufficient information and ontological choices about the proposed ontology (but recollect from Section 4.3 that one does not have to answer all questions to obtain a selection). All participants from Group B who had provided feedback on ONSET felt that they would not have been able to perform the task easily without ONSET and they agreed that the user-interface and navigation through the program was quick and simple.

Further, the alpha version of ONSET was presented informally at MAIS'11, which resulted in positive feedback, including usage of the tool also in an ontology engineering course at the University of South Africa (UNISA). The suggestion to implement a tooltip to explain relatively complicated ontological terms was implemented with the more comprehensive “explain” buttons. Positive feedback was received also from the DOLCE, BFO, and GFO ontology developers, in particular regarding the enhanced possibility for comparison with other ontologies and how a scenario has an effect on the selection, which opens avenues for further investigation.

5.3 Discussion

As the evaluation of ONSET demonstrates, it effectively lets the user select a FO and provides an explanation why. It being software-based, this makes it easy to run alternative scenarios and obtain selections, compared to the pre-existing situation with manual assessments of paper-based comparisons and where the developer had to read through all documentation before being able to make an informed selection. The method proposed here can be used at the start of ontology development, during improvement of an existing ontology with a FO, and for ontology interoperability scenarios.

It raises also some new questions and sheds light on existing ones. First, if the “FO library” envisioned in 2003 by [3] would have existed, selection of a FO would have been less of an issue, for their formalisations would have been aligned to the extent possible. Such a hypothetical library, however, would still need some management of, among others, modularity, availability of the ontology in a preferred, or deemed necessary, ontology language, and ontological choices. Hence, a library with mappings between concepts/universals and relationships/properties of the ontologies alone would not satisfy the ontology developers’ needs. Second, the whole notion of ‘foundational’ is based on the assumption that there is one and that that ontology fits for all domain ontology development projects. Yet, when one runs different scenarios, conflicting answers may arise to the extent that there may well be no ‘best fit’, i.e., where more than one FO fits equally well (or badly) given the user input provided, or at least having to deal with minor conflicting answers (recollect Fig. 3), or another is proposed due to the

given scaling of the categories. It is exactly here that the explanations become crucial: they are fact-based arguments for and against a particular FO for that scenario, and therewith compose a start for a scientific analysis of FO choice in domain ontology development projects.

While we evaluated several existing domain ontology projects, this is biassed toward the criteria described informally in the corresponding paper, hence geared toward confirmation of correct implementation of the ONSET algorithm; the selection may or may not be the same once the developers are offered the additional criteria available in ONSET. Furthermore, it would be interesting to know whether some category of criteria, or individual criteria, are always deemed more important than others, whether there exists one or more ‘typical’ combinations of criteria, and the incidence of conflicts and if so, which criteria they typically involve. ONSET clearly can be a useful aid investigating these questions, but answering them is left to future works.

6 Conclusions

The problem that ontology developers have severe difficulties in selecting which foundational ontology to use for domain ontology development and why, has been successfully solved with the ONSET tool. ONSET assists and informs developers about the criteria and properties associated with foundational ontologies and how they relate to the domain ontology to be created. It calculates a preferred foundational ontology based on the user-provided requirements and the values of the criteria for each foundational ontology. The compiled lists of criteria and implementation is, to the best of our knowledge, the first paper-based as well as software-assisted and subject-domain independent approach in foundational ontology selection. Effectiveness of ONSET was experimentally evaluated and shown to substantially improve selection and the user’s capability to motivate why.

Future works pertain to extending functionalities of ONSET, such as allowing users to map their existing ontologies to a foundational ontology, and integrating foundational ontology selection and usage in existing ontology methodologies.

Acknowledgements We wish to thank Pierre Grenon, Stefano Borgo, Claudio Masolo, Frank Loebe and Heinrich Herre for their feedback.

References

1. Keet, C.M.: The use of foundational ontologies in ontology development: an empirical assessment. In: Proc. of ESWC’11. Volume 6643 of LNCS., Springer (2011) 321–335 Heraklion, Crete, Greece, 29 May–2 June, 2011.
2. Borgo, S., Lesmo, L.: The attractiveness of foundational ontologies in industry. In: Proc. of FOMI’08, Amsterdam, The Netherlands, IOS Press (2008) 1–9
3. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A.: Ontology library. WonderWeb Deliverable D18 (ver. 1.0, 31-12-2003). (2003) <http://wonderweb.semanticweb.org>.

4. Herre, H.: General Formal Ontology (GFO): A foundational ontology for conceptual modelling. In Poli, R., Healy, M., Kameas, A., eds.: *Theory and Applications of Ontology: Computer Applications*. Springer, Heidelberg (2010) 297–345
5. Niles, I., Pease, A.: Towards a standard upper ontology. In: *Proc. of FOIS'01*, IOS Press (2001) Ogunquit, Maine, October 17-19, 2001.
6. Kassel, G.: Integration of the DOLCE top-level ontology into the OntoSpec methodology. Technical Report HAL : hal-00012203/arXiv : cs.AI/0510050, Laboratoire de Recherche en Informatique d'Amiens (LaRIA) (October 2005) <http://hal.archives-ouvertes.fr/ccsd-00012203>.
7. Grenon, P., Smith, B., Goldberg, L.: Biodynamic ontology: Applying BFO in the biomedical domain. In: *Stud. Health Technol. Inform*, IOS Press (2004) 20–38
8. Ermolayev, V., Keberle, N., Matzke, W.E.: An upper level ontological model for engineering design performance domain. In: *Proc. of ER'08*. Volume 5231 of LNCS., Springer (2008) 98–113
9. Oberle, D.: *Semantic Management of Middleware*. Volume 1 of *Semantic Web and Beyond*. Springer, New York (2006)
10. Jonquet, C., Musen, M.A., Shah, N.H.: Building a biomedical ontology recommender web service. *Journal of Biomedical Semantics* **1**(Suppl 1) (2010) S1
11. Martínez-Romero, M., Vázquez-Naya, J.M., Pereira, J., Pazos, A.: A multi-criteria approach for automatic ontology recommendation using collective knowledge. In: *Recommender Systems for the Social Web*. Volume 32 of ISRL., Springer (2012) 89–103
12. Wang, X., Guo, L., Fang, J.: Automated ontology selection based on description logic. In: *Proc. of CSCWD'08*, IEEE Computer Society (2008) 482–487 April 16-18, 2008, Xi'an, China.
13. Mascardi, V., Cord, V., Rosso, P.: A comparison of upper ontologies. Technical Report DISI-TR-06-21, University of Genova, Italy (2007) 55-64.
14. Jureta, I.J., Mylopoulos, J., Faulkner, S.: A core ontology for requirements. *Applied Ontology* **4**(3-4) (August 2009) 169–244
15. Semy, S.K., Pulvermacher, M.K., Obrst, L.J.: Toward the use of an upper ontology for us government and us military domains: An evaluation. Technical Report MTR 04B0000063, The MITRE Corporation (2004)
16. Keet, C.M.: Transforming semi-structured life science diagrams into meaningful domain ontologies with DiDOn. *J. of Biomedical Informatics* **45**(3) (2012) 482–494
17. Hoehndorf, R., Prüfer, K., Backhaus, M., Visagie, J., Kelso, J.: The design of a wiki-based curation system for the ontology of functions. In: *Proc. of the Joint BioLINK and 9th Bio-Ontologies Meeting*. (2006) Fortaleza, Brazil, Aug 5.
18. Smith, B., et al.: The OBO Foundry: Coordinated evolution of ontologies to support biomedical data integration. *Nature Biotechnology* **25**(11) (2007) 1251–1255
19. Ahmad, F., Lindgren, H.: Selection of foundational ontology for collaborative knowledge modeling in healthcare domain. In: *Proc. of AIMSA'10*. Volume 6304 of LNCS., Springer (2010) 261–262
20. Keet, C.M.: Factors affecting ontology development in ecology. In: *Proc. of DILS'05*. Volume 3615 of LNBI., Springer Verlag (2005) 46–62 San Diego, USA, 20-22 July 2005.
21. Sinha, G., Mark, D.: Toward a foundational ontology of the landscape. In: *Proc. of GIScience'10*. (2010) Zürich, Switzerland, September 14-17, 2010.
22. Fernández-López, M., Gómez-Pérez, A., Suárez-Figueroa, M.C.: Selecting and customizing a mereology ontology for its reuse in a pharmaceutical product ontology. In Eschenbach, C., Grüninger, M., eds.: *Proc. of FOIS'08*, IOS Press (2008) 181–194